UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/763,528	01/23/2004	Scott Joseph Colbeck	SJO920030100US1	9208

45216 7590 01/21/2009

Kunzler & McKenzie
8 EAST BROADWAY
SUITE 600
SALT LAKE CITY, UT 84111

EXAMINER	
FEARER, MARK D	

ART UNIT	PAPER NUMBER
2443	

MAIL DATE	DELIVERY MODE
01/21/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/763,528	COLBECK ET AL.	
	Examiner	Art Unit	
	MARK D. FEARER	2443	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 October 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-6,8-19,21-26 and 28-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-6, 8-19, 21-26 and 28-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Applicant's Amendment filed 30 October 2008 is acknowledged.

2. Claim 1 has been amended.

3. Claims 7, 20 and 27 are cancelled.

4. Claims 1-6, 8-19, 21-26 and 28-30 are pending in the present application.

5. This action is made FINAL.

Claim Rejections - 35 USC § 102

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

7. Claims 1 and 8-10 are rejected under 35 U.S.C. 102(e) as being anticipated by Anderson et al. (US 20040015566 A1).

Consider claim 1. Anderson et al. discloses an apparatus for managing data in a grid computing environment (paragraph 0010), the apparatus comprising: a GUI

generation module configured to examine user credentials and generate web-based graphical user interfaces to view data replications on multiple network nodes and enable a user to initiate modification or deletion of data on a selected node (paragraph 0563); a replication management module configured to conduct data replication operations (paragraph 0074) including directory-based replication operations based on user input (paragraph 0278); and the replication management module further configured to invoke generation of at least one graphical user interface, the at least one graphical user interface (paragraph 0565) configured to facilitate invocation of the data replication operations by a user (paragraphs 0074 and 0278).

Consider claim 8, as applied to claim 1. Anderson et al. discloses an apparatus wherein the replication operations are conducted on search results (paragraph 0241).

Consider claim 9, as applied to claim 1. Anderson et al. discloses an apparatus wherein the replication management module is further configured to change attributes associated with a file (paragraph 0188).

Consider claim 10, as applied to claim 1. Anderson et al. discloses an apparatus wherein the replication management module is further configured to conduct publishing operations (paragraph 0078).

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

9. Claims 2-4 are rejected under 35 U.S.C. 103(a) as being unpatentable over Anderson et al. (US 20040015566 A1) in view of Midgley et al. (US 20030074378 A1).

Consider claims 2-4. Anderson et al. discloses a manageable collaborative computing system. However, Anderson et al. fails to disclose of a collaborative computing system further comprising a replica location service, at least one replica location index, and at least one local replica catalog. Midgley et al. discloses a replicating system comprising a catalog process that is capable of recording metadata representing the locations of the versions of the target files on the storage medium that creates an index for accessing the versions of a target file ((“In a further aspect, the

systems and methods described herein can include backup systems that include a long term storage system for recording target data files to a storage medium in response to the operation of the dynamic replication process, thereby storing versions of the target file on the storage medium. Additionally, in an optional environment the systems may include a catalog process that can record metadata that is representative of the locations of the versions of the target files on the storage medium to thereby create an index for accessing these versions of the target file. The catalog process may include a mechanism for storing security metadata that is associated with the different versions of the target data files and that is representative of the users access rights for the versions of the target data file.") paragraph 0021).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate a replicating system comprising a catalog process that is capable of recording metadata representing the locations of the versions of the target files on the storage medium that creates an index for accessing the versions of a target file as taught by Midgley et al. with a manageable collaborative computing system as taught by Anderson et al. for the purpose of web-based grid computing comprising remote sites.

10. Claims 5-6 are rejected under 35 U.S.C. 103(a) as being unpatentable over Anderson et al. (US 20040015566 A1) in view of Zhang et al. (US 20050120353 A1).

Consider claims 5-6. Anderson et al. discloses a manageable collaborative computing system. However, Anderson et al. fails to disclose a grid system comprising a file transfer service. Zhang et al. discloses a data replication system consisting of ftp, grid ftp, http, rft, and file transfer ((“The action manager 12, which is the engine of the framework, receives collaborative messages (or CxP) messages from a design partner side, which can be a Web portal. In each message, it contains meta data or annotations describing the documents to be exchanged, such as the file name, size, author, application to use to open the design file, etc. In addition, annotations can also specify integration activities to be performed, representing new application to be integrated, such as FTP, reliable file transfer (RFT) or an invocation to a legacy adaptor. Also, an alternative data source to the Action Manager, in addition to collaborative messages, is an RDF string.”) paragraph 0024 (“The logical structure of ActivityChain ontology is shown in FIG. 3. The top-level entity is Class Activity. It has a DataTypeProperty securityHandler and an ObjectProperty actname. The ObjectProperty actname has a range which is Class Actname. And Actname is a collection which enumerates GridFTP,FTP, HTTP, Inv-service, Inv-Appl and Search-Annt.”) paragraph 0045).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate a data replication system consisting of ftp, grid ftp, http, rft, and file transfer as taught by Zhang et al. with a manageable collaborative computing system as taught by Anderson et al. for the purpose of reliable file transfer in a collaborative environment.

11. Claims 11-13, 16-18, and 24 and 8-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Midgley et al. (US 20030074378 A1) in view of Tripp (US 20050015466 A1) and in further view of Anderson et al. (US 20040015566 A1).

Consider claims 11, 16 and 24. Midgley et al. discloses a method for managing data in a grid computing environment, the method comprising: providing a graphical user interface configured to facilitate invocation of data replication operations by a user including directory-based replication operations (("The replicated data structure 54 also provides directories, subdirectories and data records.") paragraph 0040); invoking a replica location service associated with a grid (("The database can include pointers to the location of the different versions of the target files on the tape, thereby providing more rapid access to the location on the tape that includes the information a user may want to restore.") paragraph 0041); and conducting the data replication operations in response to selections on the graphical user interface by the user (("This system can provide a user interface that will allow the user to select a network consumption limit that is representative of the users selected limit for the amount of network bandwidth to be allocated to the backup replication process and the agent process.") paragraph 0019). However, Midgley et al. fails to disclose a method comprising a dynamic web based graphical user interface, a search interface for discovering replicated data including directory based replicated data, a replica location service configured to aggregate information about local replica catalogs and map logical file names to physical file names, or conduct local or remote data replication and mapping in

response to user selections. Tripp discloses a method of peer-to-peer automated anonymous asynchronous file sharing comprising a dynamic web based graphical user interface (paragraph 0081), a search interface for discovering replicated data including directory based replicated data (paragraphs 0029-0030), a replica location service configured to aggregate information about local replica catalogs and map logical file names to physical file names (paragraphs 0069 and 0083), and conduct local or remote data replication and mapping in response to user selections (paragraph 0084). Therefore, it would have been obvious for a person of ordinary skill in the art at the time the invention was made to incorporate a method of peer-to-peer automated anonymous asynchronous file sharing comprising a dynamic web based graphical user interface, a search interface for discovering replicated data including directory based replicated data, a replica location service configured to aggregate information about local replica catalogs and map logical file names to physical file names, and conduct local or remote data replication and mapping in response to user selections as taught by Tripp with a method for managing data in a grid computing environment, the method comprising: providing a graphical user interface configured to facilitate invocation of data replication operations by a user including directory-based replication operations; invoking a replica location service associated with a grid; and conducting the data replication operations in response to selections on the graphical user interface by the user as taught by Midgley et al. for the purpose of data retrieval services. However, Midgley et al., as modified by Tripp, fails to disclose a method comprising enabling a user to initiate modification or deletion of data on a selected node. Anderson et al. discloses an electronic item

management and archival system and method of operating the same comprising enabling a user to initiate modification or deletion of data on a selected node (paragraph 0563).

Therefore, it would have been obvious for a person of ordinary skill in the art at the time the invention was made to incorporate an electronic item management and archival system and method of operating the same comprising enabling a user to initiate modification or deletion of data on a selected node as taught by Anderson et al. with a method of peer-to-peer automated anonymous asynchronous file sharing comprising a dynamic web based graphical user interface, a search interface for discovering replicated data including directory based replicated data, a replica location service configured to aggregate information about local replica catalogs and map logical file names to physical file names, and conduct local or remote data replication and mapping in response to user selections and a method for managing data in a grid computing environment, the method comprising: providing a graphical user interface configured to facilitate invocation of data replication operations by a user including directory-based replication operations; invoking a replica location service associated with a grid; and conducting the data replication operations in response to selections on the graphical user interface by the user as taught by Midgley et al., as modified by Tripp, for the purpose of user initiated data retrieval services.

Consider claims 12 and 17. Midgley et al., as modified by Tripp and Anderson et al., further discloses a method comprising accessing at least one replica location index ((“Additionally, in an optional environment the systems may include a catalog process

that can record metadata that is representative of the locations of the versions of the target files on the storage medium to thereby create an index for accessing these versions of the target file.") Midgley et al., paragraph 0021).

Consider claims 13 and 18. Midgley et al., as modified by Tripp and Anderson et al., further discloses a method comprising accessing at least one local replica catalog (("The catalog process may include a mechanism for storing security metadata that is associated with the different versions of the target data files and that is representative of the users access rights for the versions of the target data file.") Midgley et al., paragraph 0021).

12. Claims 14 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Midgley et al. (US 20030074378 A1) in view of Tripp (US 20050015466 A1) in further view of Anderson et al. (US 20040015566 A1) and in further view of Zhang et al. (US 20050120353 A1).

Consider claims 14 and 19. Midgley et al., as modified by Tripp and Anderson et al., discloses a system for backing up data files comprising data replication. However, Midgley et al., as modified by Tripp, fails to disclose a system comprising a file transfer service. Zhang et al. discloses a system comprising the file transfer protocols FTP and RFT (("The action manager 12, which is the engine of the framework, receives collaborative messages (or CxP) messages from a design partner side, which can be a

Web portal. In each message, it contains meta data or annotations describing the documents to be exchanged, such as the file name, size, author, application to use to open the design file, etc. In addition, annotations can also specify integration activities to be performed, representing new application to be integrated, such as FTP, reliable file transfer (RFT) or an invocation to an legacy adaptor. Also, an alternative data source to the Action Manager, in addition to collaborative messages, is an RDF string.") paragraph 0024).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate a system comprising the file transfer protocols FTP and RFT as taught by Zhang et al. with a system comprising data replication as taught by Midgley et al., as modified by Tripp and Anderson et al., for the purpose of file transfer in a grid environment.

13. Claim 15 is rejected under 35 U.S.C. 103(a) as being unpatentable over Midgley et al. (US 20030074378 A1) in view of Tripp (US 20050015466 A1) in further view of Anderson et al. (US 20040015566 A1) and in further view of Flanagan et al. (US 6243737 B1).

Consider claim 15. Midgley et al., as modified by Tripp and Anderson et al., discloses a system for backing up data files comprising data replication. However, Midgley et al., as modified by Tripp and Anderson et al., fails to disclose a system

wherein a graphical user interface comprises a web page. Flanagan et al. discloses an interactive web-based solution (("There have been various proposed methods for providing information residing on a host system to customers through the Internet, in particular, using the Web. A typical solution involves adding new software code on the host system that interfaces with Web-based users.") column 1 lines 29-33).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate an interactive web-based solution as taught by Flanagan et al. with a system comprising data replication as taught by Midgley et al., as modified by Tripp and Anderson et al., for the purpose of inter-operability in a collaborative environment.

14. Claims 21-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Midgley et al. (US 20030074378 A1) in view of Tripp (US 20050015466 A1) in further view of Anderson et al. (US 20040015566 A1) and in further view of Wolff (US 6886035 B2).

Consider claims 21-23. Midgley et al., as modified by Tripp and Anderson et al., discloses a system for backing up data files comprising data replication. However, Midgley et al., as modified by Tripp and Anderson et al., fails to disclose a data replication system comprising replications operations that are conducted on catalog search results, a method for changing file attributes, or a publishing function. Wolff

discloses a client-server system comprising data replication further comprising volume tables constructed from a previous search (("'Total_tables 14xx' This value indicates the total number of Volume tables that have been configured and found at a previous search. This is the number that will automatically be expected to be found upon net startup.") column 62 lines 12-16), a function for changing file attributes (("Control is then passed to process 1370 where commands to get attributes of a file are managed by the metadata server. Control is then passed to process 1372 where commands to set the attributes of a file are managed by the metadata server.") column 53 lines 13-17), and a publishing function of the replicated data (("These results are replicated to each servers copy of the dynamic RAM resident configuration database 120A2-B2, the results are published and received by processes 104PC on server 104C, and the lock 120D1 is removed.") column 8 lines 25-29).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate a data replication system comprising replications operations that are conducted on catalog search results, a method for changing file attributes, and a publishing function as taught by Wolff with a data replication system as taught by Midgley et al., as modified by Tripp and Anderson et al., for the purpose of metadata catalog services in a grid computing environment.

15. Claims 25-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Midgley et al. (US 20030074378 A1) in view of James (US 6910038 B1).

Consider claims 25-26. Midgley et al. discloses a replication server configured to generate at least one graphical user interface and conduct data replication operations including directory-based replication operations in response to user selections on the graphical user interface (("This system can provide a user interface that will allow the user to select a network consumption limit that is representative of the users selected limit for the amount of network bandwidth to be allocated to the backup replication process and the agent process.") paragraph 0019 ("The replicated data structure 54 also provides directories, subdirectories and data records.") paragraph 0040). However, Midgley et al. fails to disclose a replication server comprising a replica location index. James discloses a method for host processing comprising a computing node having a replica location index, the replica location index configured to map logical names to a local replica catalog (("One embodiment of a record data structure is illustrated in FIG. 2B. Typical data fields include, as illustrated, the file parent of the data file. This information is used to map the file path to the data file in its destination location in order to locate the file on the destination CD. The volume label index is additional location identification information naming the source volume of the data file. The file size identifies the exact size of the file in bytes (or other suitable units of measure) to be used in calculating and identifying the destination location of the data and in making the determination which files will be sent to system cache memory during the writing operation. Files that are sent to system cache memory are further identified by the location in the system cache memory which holds the data file as

described in greater detail below, and the file size is used to calculate that location. The logical block number identifies the destination location by the logical block where the data file will be written. The file time is the most recent modification time of the data file. This provides both the time and the date of the file, and can be used, for example, in both cataloging as well as differentiating between two identically named files. The file source path is the complete path to the data file in order to locate and read the file during the recording operation, and the file attributes include such information as whether the file is a system file, a read-only file, if it is a hidden file, and whether it is an archive file.") column 5 lines 3-27).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate a method for host processing comprising a computing node having a replica location index, the replica location index configured to map logical names to a local replica catalog as taught by James with a replication server configured to generate at least one graphical user interface and conduct data replication operations including directory-based replication operations in response to user selections on the graphical user interface as taught by Midgley et al. for the purpose of grid hosting.

Consider claim 29. Midgley et al., as modified by James, discloses a system wherein a replication server is configured to invoke a replica location service associated with a grid (("The database can include pointers to the location of the different versions of the target files on the tape, thereby providing more rapid access to the location on the tape that includes the information a user may want to restore.") paragraph 0041).

Consider claim 30. Midgley et al., as modified by James, discloses a system wherein a replication server is configured to access at least one replica location index (("Additionally, in an optional environment the systems may include a catalog process that can record metadata that is representative of the locations of the versions of the target files on the storage medium to thereby create an index for accessing these versions of the target file.") paragraph 0021).

16. Claim 28 is rejected under 35 U.S.C. 103(a) as being unpatentable over Midgley et al. (US 20030074378 A1) in view of James (US 6910038 B1) and in further view of Wolff (US 6886035 B2).

Consider claim 28. Midgley et al., as modified by James, discloses a system for backing up data files comprising host processing methods. However, Midgley et al., as modified by James, fails to disclose a system comprising a replication server configured to conduct publishing operations, replication operations on search results, and change attributes associated with a file. Wolff discloses a system wherein a replication server is configured to conduct publishing operations (("These results are replicated to each servers copy of the dynamic RAM resident configuration database 120A2-B2, the results are published and received by processes 104PC on server 104C, and the lock 120D1 is removed.") column 8 lines 25-29), conduct replication operations on search results (("'Total_tables 14xx' This value indicates the total number of Volume tables that have been configured and found at a previous search. This is the number that will

automatically be expected to be found upon net startup.") column 62 lines 12-16), and change attributes associated with a file (("Control is then passed to process 1370 where commands to get attributes of a file are managed by the metadata server. Control is then passed to process 1372 where commands to set the attributes of a file are managed by the metadata server.") column 53 lines 13-17).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate a system wherein a replication server is configured to conduct publishing operations, conduct replication operations on search results, and change attributes associated with a file as taught by Wolff with a system for backing up data files comprising host processing methods as taught by Midgley et al., as modified by James, for the purpose of a grid data mirroring package.

Response to Arguments

17. Applicant's arguments filed 30 October 2008 with respect to claims 1, 11, 16 and 24 have been considered but are not persuasive.

Applicant argues that the Office rejected Claims 1 and 8-10 under 35 U.S.C. 102(e) as being anticipated by Anderson. It is well settled that under 35 U.S.C. § 102 "an invention is anticipated if... all the claim limitations [are] shown in a single prior art reference. Every element of the claimed invention must be literally present, arranged as in the claim. The identical invention must be shown in as complete detail as is contained in the patent claim." Richardson v. Suzuki Motor Co., Ltd., 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir.

1989). The following points of recognition will traverse the rejection by showing that every element of Applicants' invention is not present in Anderson.

The Office Action asserts that Anderson discloses a GUI generation module "configured to examine user credentials and generate web-based graphical user interfaces to view data replications on multiple network nodes and enable a user to initiate modification or deletion of data on a selected node." Anderson's GUI generation module's capability for allowing a user to view data is limited to attributes specific to an image, video, or music file. Specifically, Anderson references viewing the type of file and then being able to sort, if there are multiple results from a search. (Table 77, Anderson). Anderson discloses similar functionality for archived image, video, or music files. Anderson does not provide viewing of replicated data such as number of replications, location of replica, or time of replication. In contrast, Applicants' invention provides node-specific management of data in a grid computing environment.

Additionally, Applicants' invention provides for user input in regards to replication. That is, the user can identify the source and destination of a replication. Anderson is limited to the replication service incorporated into the CORBA bus. Anderson does not disclose obtaining information regarding archived data on a selected node, nor does it disclose invoking replications based on user input. While Anderson discloses invoking a generic replication service, Anderson does not require user input other than invoking the service itself. Applicants' invention invokes a replication service, which is not limited to the service supplied by the CORBA bus. The graphical user interface of Applicants' invention receives user input and passes that input to the replication service when it is

invoked to identify what to replicate and where to replicate it.

Therefore, Applicants' respectfully assert that Anderson does not anticipate Applicants' Claim 1 because although both inventions supply a graphical user interface, the functionality and utility of the GUI is different. Anderson does not disclose a GUI with replication management tools to initiate modification or deletion of data on a selected node as specified in Applicants' claims. Consequently, Anderson does not anticipate Claim 1 and Claims 2-10 that depend therefrom.

Examiner respectfully disagrees. Anderson et al. discloses a GUI generation module configured to examine user credentials (("To begin logging on to the EIMA system 100 (specifically the host server 110), the administrator enters the address of the EIMA Web site in the Address bar of the Web browser. After the login information has been authenticated, the administrator is able to access System Administration and any other EIMA applications for which the administrator has been granted permissions to use. The administrator must have administrator capabilities to use System Administration.") paragraph 0563), and generate web-based graphical user interfaces to view data replications on multiple network nodes (("Creation of a distributed computing system based on the CORBA standard can generally begin with an outline of desired functionality and translation of the design into software objects. The objects are expressed in terms of Interface Definition Language ("IDL") interfaces and collected into related modules. In one embodiment of the invention, the IDL is utilized for creation of

Application Programming Interface ("API") definitions that define how the client and host server systems communicate. One or more IDL files are compiled to generate stub code and skeleton code. The stub code becomes the interface that client applications use to initiate an operation from a server and is programming language independent. The skeleton code provides the interface to object implementations that the host and/or virtual servers may provide. Libraries provided through the IDL compilation provide the mechanism for communication between client and host server processes. The CORBA specification ensures that this communication be platform and language independent. Host and/or virtual server applications are created for publishing the object references by name through a naming service and, upon the request of a client application, a reference to a generic CORBA object is returned. This object reference is then narrowed to the stub representation of the remote CORBA object. The client can then invoke operations through the stub reference as if the object was local to the client. Requested operations from the client application are sent to the skeleton reference obtained through the naming service. Using the ORB, CORBA IDL stubs and skeletons serve as a connection between client and server application threads. In addition, each client and server can have threading definitions defined in a Portable Object Adapter (POA), which controls the communication to a CORBA Object by associating the object with the ORB. Each POA service may use single threaded or multiple-threaded communication protocols and the multiple-threaded protocols may be further defined as "pools" of threads or as a thread per client. The machine independence of the CORBA standard, as utilized by embodiments of the invention, allows for multiple processes to

communicate across machines, platforms, and languages, thereby providing a distributed computing environment. In another embodiment, the CORBA communication protocols are utilized to abstract client and server interactions. Using the IDL, APIs are created that separate the architecture logic. Therefore, the CORBA communication layer acts to "hide" or "mask" the host and virtual servers from the client or business logic. Each server process in the EIMA system 100 can be defined to utilize the multiple-threaded "pools" of threads, thereby allowing non-blocking calls to be handled from a large number of client applications. Each client application can be handled independently and, therefore, do not block each other during communication with the servers. The name service and event service, defined by the CORBA specification, are used to handle name lookups for services and event routing or channeling. In addition, a host or virtual server may execute multiple Generic-Input Applications ("GIAs"), statement prints, and exports at the same time, all of which may execute independent of each other and interact separately with an archive or database. Implementation of the CORBA bus in this embodiment also includes providing object services for use by multiple distributed object programs. These services include domain-independent interfaces such as the naming service, a trading service, a repository service, an indexing service, a set service, a parameter service, a log service, an application service, and a redundancy/replication service. The services are available to CORBA objects and a client may initiate multiple services if desired. For example, a client application may invoke multiple services when interfacing data with input/output ("I/O") devices. Alternatively, multiple threads can exist within the services themselves. For

example, depending on the operation, a user or client may invoke multiple threads within the repository service. In some embodiments, the EIMA system may also implement a factory concept whereby a server is a service "factory" that handles queries each time a client connects and requests an individual session. Each session manages its own client and then allows for the abstraction and separation of logic for multiple client applications. “) paragraphs 0078-0079), and enable a user to initiate modification or deletion of data on a selected node ((“The user then selects the Delete Database option 715. The user can then select a database to delete. That database is deleted from the archive. c. Migrating from RAID to Optical Migrating from RAID to optical allows the user to move a copy of the document data and image files in a particular database cycle from disk (RAID) to an optical device. The migration process enables the user to free up disk space. After the user has successfully migrated the files from disk to optical, the user can delete the database files from disk. The Optical Repository server and the Optical Robotic server need to be running. To migrate images from RAID to Optical, the user performs the following acts: At the Main Menu 500, the user checks that the correct database and cycle that he wants to migrate to RAID are selected. The user then selects the File Management & Utilities Menu 520 option. At the File Management & Utilities Menu 700, the user selects the Start Migration option. The user then chooses the source (e.g., RAID repository) and the destination. Once the source and destination are selected, the migration is performed.”) paragraphs 0318-0322); a replication management module configured to conduct data replication operations including directory-based replication operations based on user

input ((“A host machine for an electronic item management and archival system, the host machine comprising: a communications endpoint that receives items, each item including at least one of image data, audio data, and video data; a processor connected to the communications endpoint; and software executable by the processor, the software including instructions that create one or more virtual servers, the one or more virtual servers including at least one server that facilitates independent communication between multiple Common Object Request Broker Architecture (CORBA) applications and at least one server that creates and manages an archive.”) Claim 21 (“Once the import process is initialized, the application is in an "In Process" mode and the screen displays the XML filename, date and time of the import process and the number of documents that are being imported to EIMA system 100. To stop the import process or clear the display, the user clicks the stop or clear list buttons 4305 or 4310. When all the files have been imported to the database, the application continues to check the source directory and imports documents that are available from that directory. This window can be minimized so the import process application runs in the background.”) paragraph 1102); and the replication management module further configured to invoke generation of at least one graphical user interface, the at least one graphical user interface configured to facilitate invocation of the data replication operations by a user ((“A host machine for an electronic item management and archival system, the host machine comprising: a communications endpoint that receives items, each item including at least one of image data, audio data, and video data; a processor connected to the communications endpoint; and software executable by the processor, the software

including instructions that create one or more virtual servers, the one or more virtual servers including at least one server that facilitates independent communication between multiple Common Object Request Broker Architecture (CORBA) applications and at least one server that creates and manages an archive.") Claim 21 ("Creation of a distributed computing system based on the CORBA standard can generally begin with an outline of desired functionality and translation of the design into software objects. The objects are expressed in terms of Interface Definition Language ("IDL") interfaces and collected into related modules. In one embodiment of the invention, the IDL is utilized for creation of Application Programming Interface ("API") definitions that define how the client and host server systems communicate. One or more IDL files are compiled to generate stub code and skeleton code. The stub code becomes the interface that client applications use to initiate an operation from a server and is programming language independent. The skeleton code provides the interface to object implementations that the host and/or virtual servers may provide. Libraries provided through the IDL compilation provide the mechanism for communication between client and host server processes. The CORBA specification ensures that this communication be platform and language independent. Host and/or virtual server applications are created for publishing the object references by name through a naming service and, upon the request of a client application, a reference to a generic CORBA object is returned. This object reference is then narrowed to the stub representation of the remote CORBA object. The client can then invoke operations through the stub reference as if the object was local to the client. Requested operations from the client

application are sent to the skeleton reference obtained through the naming service. Using the ORB, CORBA IDL stubs and skeletons serve as a connection between client and server application threads. In addition, each client and server can have threading definitions defined in a Portable Object Adapter (POA), which controls the communication to a CORBA Object by associating the object with the ORB. Each POA service may use single threaded or multiple-threaded communication protocols and the multiple-threaded protocols may be further defined as "pools" of threads or as a thread per client. The machine independence of the CORBA standard, as utilized by embodiments of the invention, allows for multiple processes to communicate across machines, platforms, and languages, thereby providing a distributed computing environment. In another embodiment, the CORBA communication protocols are utilized to abstract client and server interactions. Using the IDL, APIs are created that separate the architecture logic. Therefore, the CORBA communication layer acts to "hide" or "mask" the host and virtual servers from the client or business logic. Each server process in the EIMA system 100 can be defined to utilize the multiple-threaded "pools" of threads, thereby allowing non-blocking calls to be handled from a large number of client applications. Each client application can be handled independently and, therefore, do not block each other during communication with the servers. The name service and event service, defined by the CORBA specification, are used to handle name lookups for services and event routing or channeling. In addition, a host or virtual server may execute multiple Generic-Input Applications ("GIAs"), statement prints, and exports at the same time, all of which may execute independent of each other and interact

separately with an archive or database. Implementation of the CORBA bus in this embodiment also includes providing object services for use by multiple distributed object programs. These services include domain-independent interfaces such as the naming service, a trading service, a repository service, an indexing service, a set service, a parameter service, a log service, an application service, and a redundancy/replication service. The services are available to CORBA objects and a client may initiate multiple services if desired. For example, a client application may invoke multiple services when interfacing data with input/output ("I/O") devices. Alternatively, multiple threads can exist within the services themselves. For example, depending on the operation, a user or client may invoke multiple threads within the repository service. In some embodiments, the EIMA system may also implement a factory concept whereby a server is a service "factory" that handles queries each time a client connects and requests an individual session. Each session manages its own client and then allows for the abstraction and separation of logic for multiple client applications.") paragraphs 0078-0079).

Applicant argues that in Claims 11, 16, 24 and 25, Midgley is not directed to subject matter that will contribute useful prior art material to substantiate a rejection.
The Examiner claims that Midgley discloses "conducting the data replication operations in response to selections on the graphical user interface by the user." The GUI in Midgley is not used to conduct data replication. The GUI in Midgley is used to control "network consumption" by enabling a user to select a network consumption limit on

bandwidth. The association with replication stems from the bandwidth being consumed by a replication process. The replication process requires regulation to limit the amount of data allowed to pass through at any given time to prevent the replication process from using too much bandwidth. Midgley does not disclose a GUI with data replication management controls. Rather, Midgley discloses a GUI with bandwidth management controls that can limit the bandwidth used by processes such as replication.

Anderson is combined to modify Midgley in view of Tripp in order to make obvious enabling a user to initiate modification or deletion of data on a selected node. However, as discussed earlier with regards to the 102(e) rejection, Anderson does not disclose a GUI with replication management tools to modify or delete data on a selected node.

Claims 11, 16, and 24 were rejected under 35 U.S.C. 103(a) as being unpatentable over Midgley in view of Tripp in further view of Anderson. However, because the disclosures relied

on in Midgley and Anderson have been shown to not enable Applicants' invention, Applicants assert that Claims 11, 16, and 24 should be allowed. Claim 25 was rejected under 35 U.S.C. 103(a) as being unpatentable over Midgley in view of James. Similarly, because the disclosure relied on in Midgley has been shown to not enable Applicants' invention, Applicants assert that Claim 25 should be allowed. Applicants respectfully request that the claims depending from Claims 11, 16, and 25 also be allowed as depending from allowed claims.

Examiner respectfully disagrees. Midgley et al., as modified by Tripp and Anderson et al., discloses providing a dynamic web-based graphical user interface configured to facilitate invocation of data replication operations by a user including directory-based replication

operations ((“To begin logging on to the EIMA system 100 (specifically the host server 110), the administrator enters the address of the EIMA Web site in the Address bar of the Web browser. After the login information has been authenticated, the administrator is able to access System Administration and any other EIMA applications for which the administrator has been granted permissions to use. The administrator must have administrator capabilities to use System Administration.”) Anderson et al., paragraph 0563 (“A host machine for an electronic item management and archival system, the host machine comprising: a communications endpoint that receives items, each item including at least one of image data, audio data, and video data; a processor connected to the communications endpoint; and software executable by the processor, the software including instructions that create one or more virtual servers, the one or more virtual servers including at least one server that facilitates independent communication between multiple Common Object Request Broker Architecture (CORBA) applications and at least one server that creates and manages an archive.”) Anderson et al., Claim 21 (“Once the import process is initialized, the application is in an "In Process" mode and the screen displays the XML filename, date and time of the import process and the number of documents that are being imported to EIMA system 100. To stop the import process or clear the display, the user clicks the stop or clear list buttons 4305 or 4310.

When all the files have been imported to the database, the application continues to check the source directory and imports documents that are available from that directory. This window can be minimized so the import process application runs in the background.") Anderson et al., paragraph 1102); enabling a user to initiate modification or deletion of data on a selected node (("The user then selects the Delete Database option 715. The user can then select a database to delete. That database is deleted from the archive. c. Migrating from RAID to Optical Migrating from RAID to optical allows the user to move a copy of the document data and image files in a particular database cycle from disk (RAID) to an optical device. The migration process enables the user to free up disk space. After the user has successfully migrated the files from disk to optical, the user can delete the database files from disk. The Optical Repository server and the Optical Robotic server need to be running. To migrate images from RAID to Optical, the user performs the following acts: At the Main Menu 500, the user checks that the correct database and cycle that he wants to migrate to RAID are selected. The user then selects the File Management & Utilities Menu 520 option. At the File Management & Utilities Menu 700, the user selects the Start Migration option. The user then chooses the source (e.g., RAID repository) and the destination. Once the source and destination are selected, the migration is performed.") Anderson et al., paragraphs 0318-0322); providing a search interface for discovering replicated data including directory based replicated data (("A number of file discovery and sharing programs have become very popular for use across networks, especially those programs which permit the sharing of multimedia content. Users connect to a central directory service and

upload a list of files that they currently have on their local system which may be requested by other participants in the directory service. To retrieve files, users send a request for a file to the central directory service which then connects the requesting user to another user's computer containing that file which computer is also currently online. The most popular program of this type is Napster, a utility for sharing audio files by manually registering them with a central directory service. Another popular program is Gnutella which shares more general-purpose files. The general term for both programs is a "peer-to-peer file sharing service".") Tripp, paragraph 0002 ("In transferring data from each remote location, data is typically stored at the remote location and then transferred to and replicated at the central location. One of four methods is generally used to update the central database, as previously discussed above under the Background section. First, all remotely stored data is copied over the intranet to the central location. Second, only those files or objects that have changed since the last transfer are copied to the central location. Third, a transaction log is kept at the remote location and transmitted to the central location, and the transaction log this then applied at the central location to update the central database. Finally, at each remote location a prior copy of the local data is compared to the current copy of the local data to generate a differential record indicating changes between the prior and current copies, and this differential record is then transferred to the central location and incorporated into the central database.") Tripp, paragraph 0085 ("The components of the central server 202 and their general operation have been described, and now the operation of the agent 204 and brochure 206 will be described in more detail. The agent 204 and brochure 206

may both be present at a remote server 208. A brochure 206 and agent can function independently of each other, as will be discussed in more detail below. The agent 204 is a small local program which executes at the remote server 208 and generates an incremental search engine update for all of the participating web sites on the web host 208. These index updates are transmitted by the agent 204 to the central server 202, where they are queued for addition to the central index.") Tripp, paragraph 0032 ("The agent 204 runs on a system, such as a web host server, at the site of an organization, and processes content (objects) for all web sites available via mass storage from that system. The agent 204 processes all web sites located within the mass storage area to which it has access, unless configured to exclude some portion of a site or sites. The agent 204 uses the local web server configuration (object catalog or file system information) data to determine the root directory path (or other location information for the particular file system) for all web site file structures available. The agent 204 reads files directly from local mass storage, and indexes the keywords from the files and meta data about the files. In contrast, a spider program, as previously discussed, is located on a server remote from the local site and renders each web page file before tokenizing and parsing each page for indexing. The agent 204 follows the structure of the local mass storage directory tree in indexing the files, and does not follow uniform resource locators ("URLs") stored within the HTML files forming the web pages. Since the agent 204 is present at the remote server 208 and has access to files stored on the server's mass storage, the agent is potentially capable of retrieving non-html data for indexing from these locally stored files, such as database files and other non web-page source

material. For example, a product catalog stored in a database file on the remote mass storage may be accessed and indexed by the agent 204.") Tripp, paragraph 0033); invoking a replica location service configured to aggregate information about local replica catalogs and map logical file names to physical file names; (("In a further aspect, the systems and methods described herein can include backup systems that include a long term storage system for recording target data files to a storage medium in response to the operation of the dynamic replication process, thereby storing versions of the target file on the storage medium. Additionally, in an optional environment the systems may include a catalog process that can record metadata that is representative of the locations of the versions of the target files on the storage medium to thereby create an index for accessing these versions of the target file. The catalog process may include a mechanism for storing security metadata that is associated with the different versions of the target data files and that is representative of the users access rights for the versions of the target data file. Such information may be employed by a secure restore process for determining the access right of a party that is requesting to restore a version of a target data file, or to view a version of a target data file including the metadata such as file name, file owner, and other information, before granting the requesting party access to the stored version of the target data file.") Midgley et al., paragraph 0021 ("It will be noted that in the embodiment described above the backup server 12 is operating under the control of the Windows NT operating system and so is the data server. Accordingly, the journal files are capturing IRP packets that are in a format generally suitable for being played both on the data server and the backup

server. However, it will be understood that in alternative embodiments the data servers may be UNIX systems or systems running another type of operating system. Similarly, the backup server may also be a server running an operating system that is different from Windows NT and different from the operating system of the data server. In this embodiment, the transaction processor would include an interpreter that could map operating system or curl mode operations captured in the journal files for the data server to operating system calls or kernel operations for the operating system for the backup server. Other enhancements and modifications of the systems and methods described herein can be practiced with the invention without departing from the scope hereof.") Midgley et al., paragraph 0072); and conducting local or remote data replication and mapping operations in response to selections on the graphical user interface by the user (("At the PC workstation, the user selects the Repair GUI application.") Anderson et al., paragraph 0837 ("The systems and methods described herein provide for continuous back up of data stored on a computer network. To this end the systems of the invention include a synchronization replication process that replicates selected source data files data stored on the network to create a corresponding set of replicated data files, called the target data files, that are stored on a back up server. This synchronization replication process builds a baseline data structure of target data files. In parallel to this synchronization process, the system includes a dynamic replication process that includes a plurality of agents, each of which monitors file access operations for a server on the network to detect and capture, at the byte-level, changes to the source data files. Each agent may record the changes to a

respective journal file, and as the dynamic replication process detects that the journal files contain data, the journal files are transferred or copied to the back up server so that the captured changes can be written to the appropriate ones of the target data files.") Midgley et al., paragraph 0008 ("It will be noted that in the embodiment described above the backup server 12 is operating under the control of the Windows NT operating system and so is the data server. Accordingly, the journal files are capturing IRP packets that are in a format generally suitable for being played both on the data server and the backup server. However, it will be understood that in alternative embodiments the data servers may be UNIX systems or systems running another type of operating system. Similarly, the backup server may also be a server running an operating system that is different from Windows NT and different from the operating system of the data server. In this embodiment, the transaction processor would include an interpreter that could map operating system or curl mode operations captured in the journal files for the data server to operating system calls or kernel operations for the operating system for the backup server. Other enhancements and modifications of the systems and methods described herein can be practiced with the invention without departing from the scope hereof.") Midgley et al., paragraph 0072).

Conclusion

18. THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any response to this Office Action should be faxed to (571) 273-8300 or mailed to:

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Hand-delivered responses should be brought to
Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Mark Fearer whose telephone number is (571) 270-1770. The Examiner can normally be reached on Monday-Thursday from 7:30am to 5:00pm.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Tonia Dollinger can be reached on (571) 272-4170. The fax phone number

for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free) or 571-272-4100.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist/customer service whose telephone number is (571) 272-2600.

Mark Fearer
/M.D.F./
January 16, 2009

/George C. Neurauter, Jr./
Primary Examiner, Art Unit 2443